

Docket No. AUS920030820US1

**CLAIMS:**

What is claimed is:

1. A method, in a data processing system, for identifying differences between the execution of a first build of a computer program and a second build of a computer program, comprising:

obtaining a first call tree data structure corresponding to first trace data of an execution of the first build of the computer program;

obtaining a second call tree data structure corresponding to second trace data of an execution of the second build of the computer program;

subtracting the second call tree data structure from the first call tree data structure to generate a subtracted call tree data structure; and

outputting the subtracted call tree data structure, wherein the subtracted call tree data structure identifies differences between the execution of the first build of the computer program and the execution of the second build of the computer program.

2. The method of claim 1, further comprising:

inputting the first trace data and the second trace data to an arcflow tool, wherein the arcflow tool generates the first call tree data structure and the second call tree data structure based on the first trace data and the second trace data.

Docket No. AUS920030820US1

3. The method of claim 1, wherein the first call tree data structure and the second call tree data structure are xtree data structures.

4. The method of claim 1, wherein subtracting the second call tree data structure from the first call tree data structure to generate a subtracted call tree data structure includes:

    copying the first call tree data structure and inserting a pass field in each node of the first call tree data structure;

    initializing the pass field in each node of the first call tree data structure; and

    walking the second call tree data structure over the first call tree data structure to generate the subtracted call tree data structure.

5. The method of claim 4, wherein walking the second call tree data structure over the first call tree data structure includes:

    for each node that exists in both the first call tree data structure and the second call tree data structure, generating a node in the subtracted call tree data structure by subtracting a base value of the node in the second call tree data structure from a base value of a corresponding node in the first call tree data structure.

Docket No. AUS920030820US1

6. The method of claim 4, wherein walking the second call tree data structure over the first call tree data structure includes:

for each node that exists in only one of the first call tree data structure and the second call tree data structure, creating a node in the subtracted call tree data structure having a negative base value corresponding to a base value of the node that exists in either of the first call tree data structure or the second call tree data structure.

7. The method of claim 5, further comprising:

setting a value of a pass field of the node in the subtracted call tree data structure to a value indicating that both the first call tree data structure and the second call tree data structure contributed to the base value

8. The method of claim 6, further comprising:

setting a value of a pass field of the node in the subtracted call tree data structure to a value indicating that either the first call tree data structure or the second call tree data structure contributed to the base value.

9. The method of claim 4, wherein nodes in the first call tree data structure and nodes in the second call tree data structure whose paths have not changed between builds are not present in the subtracted call tree data structure.

Docket No. AUS920030820US1

10. The method of claim 1, further comprising:

identifying improvements or regressions from the first build to the second build of the computer program based on values associated with nodes in the subtracted call tree data structure.

11. A computer program product in a computer readable medium for identifying differences between the execution of a first build of a computer program and a second build of a computer program, comprising:

first instructions for obtaining a first call tree data structure corresponding to first trace data of an execution of the first build of the computer program;

second instructions for obtaining a second call tree data structure corresponding to second trace data of an execution of the second build of the computer program;

third instructions for subtracting the second call tree data structure from the first call tree data structure to generate a subtracted call tree data structure; and

fourth instructions for outputting the subtracted call tree data structure, wherein the subtracted call tree data structure identifies differences between the execution of the first build of the computer program and the execution of the second build of the computer program.

12. The computer program product of claim 11, further comprising:

Docket No. AUS920030820US1

fifth instructions for inputting the first trace data and the second trace data to an arcflow tool, wherein the arcflow tool generates the first call tree data structure and the second call tree data structure based on the first trace data and the second trace data.

13. The computer program product of claim 11, wherein the first call tree data structure and the second call tree data structure are xtree data structures.

14. The computer program product of claim 11, wherein the third instructions for subtracting the second call tree data structure from the first call tree data structure to generate a subtracted call tree data structure include:

instructions for copying the first call tree data structure and inserting a pass field in each node of the first call tree data structure;

instructions for initializing the pass field in each node of the first call tree data structure; and

instructions for walking the second call tree data structure over the first call tree data structure to generate the subtracted call tree data structure.

15. The computer program product of claim 14, wherein the instructions for walking the second call tree data structure over the first call tree data structure include:

for each node that exists in both the first call tree data structure and the second call tree data

Docket No. AUS920030820US1

structure, instructions for generating a node in the subtracted call tree data structure by subtracting a base value of the node in the second call tree data structure from a base value of a corresponding node in the first call tree data structure.

16. The computer program product of claim 14, wherein the instructions for walking the second call tree data structure over the first call tree data structure include:

for each node that exists in only one of the first call tree data structure and the second call tree data structure, instructions for creating a node in the subtracted call tree data structure having a negative base value corresponding to a base value of the node that exists in either of the first call tree data structure or the second call tree data structure.

17. The computer program product of claim 15, further comprising:

instructions for setting a value of a pass field of the node in the subtracted call tree data structure to a value indicating that both the first call tree data structure and the second call tree data structure contributed to the base value

18. The computer program product of claim 16, further comprising:

instructions for setting a value of a pass field of the node in the subtracted call tree data structure to a

Docket No. AUS920030820US1

value indicating that either the first call tree data structure or the second call tree data structure contributed to the base value.

19. The computer program product of claim 11, further comprising:

fifth instructions for identifying improvements or regressions from the first build to the second build of the computer program based on values associated with nodes in the subtracted call tree data structure.

20. An apparatus for identifying differences between the execution of a first build of a computer program and a second build of a computer program, comprising:

means for obtaining a first call tree data structure corresponding to first trace data of an execution of the first build of the computer program;

means for obtaining a second call tree data structure corresponding to second trace data of an execution of the second build of the computer program;

means for subtracting the second call tree data structure from the first call tree data structure to generate a subtracted call tree data structure; and

means for outputting the subtracted call tree data structure, wherein the subtracted call tree data structure identifies differences between the execution of the first build of the computer program and the execution of the second build of the computer program.